Network Working Group
INTERNET DRAFT
OBSOLETES: RFC 1960
Expire in six months

Tim Howes
Netscape Communications Corp.
March 1997

The String Representation of LDAP Search Filters

<draft-ietf-asid-ldapv3-filter-00.txt>

1. Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as ``work in progress.''

To learn the current status of any Internet-Draft, please check the ``1id-abstracts.txt'' listing contained in the Internet- Drafts Shadow Directories on ds.internic.net (US East Coast), nic.nordu.net (Europe), ftp.isi.edu (US West Coast), or munnari.oz.au (Pacific Rim).

2. Abstract

The Lightweight Directory Access Protocol (LDAP) [1] defines a network representation of a search filter transmitted to an LDAP server. Some applications may find it useful to have a common way of representing these search filters in a human-readable form. This document defines a human-readable string format for representing LDAP search filters.

This document replaces RFC 1960, extending the string LDAP filter definition to include support for LDAP version 3 extended match filters, and including support for representing the full range of possible LDAP search filters.

3. LDAP Search Filter Definition

An LDAPv3 search filter is defined in [1] as follows:

```
     Filter ::=3D CHOICE {
             and             [0] SET OF Filter,
             or              [1] SET OF Filter,
             not             [2] Filter,
```

=0C


```
            equalityMatch      [3] AttributeValueAssertion,
            substrings         [4] SubstringFilter,
            greaterOrEqual     [5] AttributeValueAssertion,
            lessOrEqual        [6] AttributeValueAssertion,
            present            [7] AttributeDescription,
            approxMatch        [8] AttributeValueAssertion,
            extensibleMatch    [9] MatchingRuleAssertion
    }

    SubstringFilter ::=3D SEQUENCE {
            type    AttributeDescription,
            SEQUENCE OF CHOICE {
                    initial        [0] LDAPString,
                    any            [1] LDAPString,
                    final          [2] LDAPString
            }
    }

    AttributeValueAssertion ::=3D SEQUENCE {
            attributeDesc   AttributeDescription,
            attributeValue  AttributeValue
    }

    MatchingRuleAssertion ::=3D SEQUENCE {
            matchingRule    [1] MatchingRuleID OPTIONAL,
            type            [2] AttributeDescription OPTIONAL,
            matchValue      [3] AssertionValue,
            dnAttributes    [4] BOOLEAN DEFAULT FALSE
    }

    AttributeDescription ::=3D LDAPString

    AttributeValue ::=3D OCTET STRING

    MatchingRuleID ::=3D LDAPString

    AssertionValue ::=3D OCTET STRING

    LDAPString ::=3D OCTET STRING
```

where the LDAPString above is limited to the UTF-8 encoding of the ISO 10646 **[4] character set.** The AttributeDescription is a string representation of the attribute description name and is defined in [1]. The AttributeValue and AssertionValue OCTET STRING have the form defined in

[2]. The Filter is encoded for transmission over a network using the Basic Encoding Rules defined in [3], with simplifications described in [1].

=0C

4. String Search Filter Definition

The string representation of an LDAP search filter is defined by the following grammar. The filter format uses a prefix notation.

```
     <filter> ::=3D '(' <filtercomp> ')'
     <filtercomp> ::=3D <and> | <or> | <not> | <item>
     <and> ::=3D '&' <filterlist>
     <or> ::=3D '|' <filterlist>
     <not> ::=3D '!' <filter>
     <filterlist> ::=3D <filter> | <filter> <filterlist>
     <item> ::=3D <simple> | <present> | <substring> | <extensible>
     <simple> ::=3D <attr> <filtertype> <value>
     <filtertype> ::=3D <equal> | <approx> | <greater> | <less>
     <equal> ::=3D '=3D'
     <approx> ::=3D '~=3D'
     <greater> ::=3D '>=3D'
     <less> ::=3D '<=3D'
     <extensible> ::=3D ( NULL | <attr> ) [ ':dn' ] [ ':' <matchingrule> =
]
                              ':=3D' <value>
     <matchingrule> ::=3D <matchingrulename> | <oid>
     <matchingrulename> ::=3D <string>
     <oid> ::=3D <string>
     <present> ::=3D <attr> '=3D*'
     <substring> ::=3D <attr> '=3D' <initial> <any> <final>
     <initial> ::=3D NULL | <value>
     <any> ::=3D '*' <starval>
     <starval> ::=3D NULL | <value> '*' <starval>
     <final> ::=3D NULL | <value>
```

<attr> is a string representing an AttributeDescription, and has the format defined in [1]. <value> is a string representing an AttributeValue, or part of one, and has the form defined in [2].

If a <value> should contain any of the characters '*' (ASCII 0x2a) or 0x00), the character must be encoded as the backslash '\' character followed by the two hexadecimal digits representing the encoded character.

This simple escaping mechanism eliminates filter-parsing ambiguities and allows the construction of any filter that can be represented in LDAP. The case of the two hexadecimal digits is not significant. Other characters besides the ones listed above may be escaped using this mechanism, for example, non-printing characters.

For example, the filter checking whether the "cn" attribute contained a value with the character "*" anywhere in it would be represented as "(cn=3D*2a*)".

=0C

Note that although both the <substring> and <present> productions can produce the 'attr=3D*' construct, this construct is used only to denote = a
presence filter.

<oid> is a dotted string representation of an object identifier (e.g., "1.2.3.4") identifying a matching rule to use when comparing <value>. <matchingrulename> is a name given to a matching rule, as defined in [2]. One of <attr> or <matchingrule> is required in the <extensible> production.

5. Examples

This section gives a few examples of search filters written using this notation.

```
(cn=3DBabs Jensen)
(!(cn=3DTim Howes))
(&(objectClass=3DPerson)(|(sn=3DJensen)(cn=3DBabs J*)))
(o=3Duniv*of*mich*)
```

The following examples illustrate the use of extensible matching.

```
(cn:1.2.3.4.5:=3DFred Flintstone)
(sn:dn:2.4.6.8.10:=3DBarney Rubble)
(o:dn:=3DAce Industry)
```

The second example illustrates the use of the ":dn" notation to indicate that matching rule "2.4.6.8.10" should be used when making comparisons, and that the attributes of an entry's distinguished name should be con-sidered part of the entry when evaluating the match.

The third example denotes an equality match, except that DN components should be considered part of the entry when doing the match.

The following examples illustrate the use of the escaping mechanism.

```
(o=3DParens R Us \28for all your parenthetical needs\29)
(cn=3D*\2A*)
(filename=3DC:\5cMyFile)
(bin=3D\00\00\00\04)
(sn=3DLu\c4\8di\c4\c7)
```

The first example shows the use of the escaping mechanism to represent

parenthesis characters. The second shows how to represent a "*" in a value, preventing it from being interpreted as a substring indicator. The third illustrates the escaping of the backslash character.

The fourth example shows a filter searching for the four-byte value

=0C

0x00000004, illustrating the use of the escaping mechanism to represent arbitrary data, including NUL characters.

The final example illustrates the use of the escaping mechanism to represent various non-printing UTF-8 characters.

6. Security Considerations

Security considerations are not discussed in this document.

7. Bibliography

[1] Lightweight Directory Access Protocol (v3), M. Wahl, T. Howes, S. Kille, Internet Draft draft-ietf-asid-ldapv3-protocol-04.txt, March, 1997.

[2] Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions, M. Wahl, A. Coulbeck, T. Howes, S. Kille, Internet Draft draft-ietf-asid-ldapv3-attributes-04.txt, March, 1997.

[3] Specification of ASN.1 encoding rules: Basic, Canonical, and Distinguished Encoding Rules, ITU-T Recommendation X.690, 1994.

[4] Universal Multiple-Octet Coded Character Set (UCS) - Architecture and Basic Multilingual Plane, ISO/ IEC 10646-1, 1993.

8. Author's Address

Tim Howes
Netscape Communications Corp.
501 E. Middlefield Road
Mountain View, CA 94043
USA
+1 415 937-3419
howes@netscape.com